1 <!--showValues---------- >

Rules

About This Section

Android & Friends

Chrome & Friends

Google & Friends

Abuse Vulnerability Reward Program Rules

Cloud Vulnerability Reward Program Rules

Google and Alphabet Vulnerability Reward Program (VRP) Rules

Open Source

Other

Program Rules

An "abuse risk" can be defined as a product feature that can cause unexpected damage to a user or platform when leveraged in an unexpected manner. Abuse risks arise when a product doesn't have sufficient guardrails in place to protect its features from being (mis)used in a malicious way.

Contrary to security vulnerabilities, where an identified loophole requires a fix, abuse risks are often inherent to product features. That means that oftentimes, features shouldn't be disabled, but that these product features require protections mitigating their exploitation at scale.

For more details on what constitutes an abuse risk, see our What is an abuse risk? article.

Scope

Any abuse risk in services under the Google VRP scope is in scope for the Abuse VRP.

Ineligible report types

When submitting AI vulnerability reports to Google's VRPs, **take note that non-qualifying issues and vulnerabilities are generally ineligible for reward**. Reports that fall in these categories will not be accepted, and your report will be closed. Examples include:

- Generating violative, misleading, or factually incorrect content within the attacker's own session (including standard "jailbreaks" and "hallucinations"). Please report inappropriate content using in-product links.
- AI-generated content-based issues, including reports of AI safety or alignment bypasses. Please report this content using in-product links.
- Compliance, legal, or intellectual property issues, including issues relating to country/region-specific laws such as privacy or intellectual property laws. Please report those issues to the Report Content for Legal Reasons process.
- Preamble extraction without sensitive information leakage.

- Contexts in which a model's incorrect output or classification does not pose a compelling attack scenario or feasible path to user harm.

Reports should be written by the reporter and demonstrate a clear in-scope threat, risk, or vulnerability in plain language. See here for the full list of non-qualifying vulnerabilities.

Qualifying vulnerabilities

Abuse-related vulnerabilities are in scope for this program if the reported attack scenario displays a design or implementation issue in a Google product that could lead to significant harm.

Some examples of abuse-related vulnerabilities:

- A technique by which an attacker is able to manipulate the rating score of a listing on Google Maps by submitting a sufficiently large volume of fake reviews that go undetected by our abuse systems. In contrast, reporting a specific business with likely fake ratings would not qualify.
- The ability to import your contacts into a social network app to see which one of your friends is using the app is considered a feature. But, this feature can become an abuse risk if there isn't a quota in place on the amount of contact lookups that can be performed within the app during a given timeframe. Without any restrictions in place, malicious actors could use this feature to build a large database of users for their spam campaigns.

See this additional information detailing how we assess abuse risk reports.

We've also detailed our criteria for bug reports on AI products – see the following section to learn what types of reports are in scope for our program.

Qualifying vulnerabilities in AI products

The following table details our criteria for AI bug reports to assist our bug hunting community in effectively testing the safety and security of our AI products. Our scope aims to facilitate testing for traditional security vulnerabilities as well as risks specific to AI systems. It is

Category	Attack Scenario	Guidance
Prompt Attacks: Crafting adversarial prompts that allow an adversary to influence the behavior of the model, and hence the output in ways that were not intended by the application	Prompt injections that are invisible to victims and change the state of the victim's account or or any of their assets.	In Scope
	Prompt injections into any tools in which the response is used to make decisions that directly affect victim users.	In Scope
	Prompt or preamble extraction in which a user is able to extract the initial prompt used to prime the model only when sensitive information is present in the extracted preamble.	In Scope
Training Data Extraction: Attacks that are able to successfully reconstruct verbatim training examples that contain sensitive information. Also called membership inference.	Training data extraction that reconstructs items used in the training data set that leak sensitive, non-public information.	In Scope
Manipulating Models: An attacker is able to covertly change the behavior of a model such that they can trigger pre-defined adversarial behaviors.	Adversarial output or behavior that an attacker can reliably trigger via specific input in a model owned and operated by Google ("backdoors"). Only in-scope when a model's output is used to change the state of a victim's account or data.	In Scope
	Attacks in which an attacker manipulates the training data of the model to influence the model's output in a victim's session according to the attacker's preference. Only in-scope when a model's output is used to change the state of a victim's account or data.	In Scope
Adversarial Perturbation: Inputs that are provided to a model that results in a deterministic, but highly	Contexts in which an adversary can reliably trigger a misclassification in a security control that can be abused for malicious use or adversarial gain.	In Scope

Model Theft / Exfiltration: AI models often include sensitive intellectual property, so we place a high priority on protecting these assets. Exfiltration attacks allow attackers to steal details about a model such as its architecture or weights.	Attacks in which the exact architecture or weights of a confidential/proprietary model are extracted.	In Scope
If you find a flaw in an AI-powered tool other than what is listed above, you can still submit, provided that it meets the qualifications listed on this page.	A bug or behavior that clearly meets our qualifications for a valid security or abuse issue.	In Scope

Non-qualifying vulnerabilities

In addition to the list of non-qualifying vulnerabilities for the Google VRP, please review the table below detailing the explicit, non-qualifying vulnerabilities for Prompt Injection techniques, which should be reported via in-product feedback channels instead of the Abuse VRP.

Common Outcome / Technique	Description / Examples
Policy-Violating Content Generation	Generating violative content (such as hate speech, violent content, or illegal instructions), misleading information, or factually incorrect/fabricated content within the user's own session (e.g. jailbreaks, safety bypasses, or alignment bypasses).
Simple Filter / Guardrail Bypass	Circumventing basic content filters or instructions (e.g., using obfuscation, translation, role-playing) to achieve policy-violating content generation (see above).
Harm Achievable Through Existing Non-AI Tools	Using an AI product to do something potentially harmful that is already possible with other tools. For example, finding a vulnerability in open source software (already possible using

Hallucinations / Factual Inaccuracy	Model generates incorrect, nonsensical, or fabricated information, including fake code execution results, APIs, or credentials.
No Compelling Harm from Incorrect Output / Classification	Contexts in which a model's incorrect output or classification does not pose a compelling attack scenario or feasible path to user harm.
Intended Functionality	Demonstrating expected behavior, e.g., accessing / summarizing the user's own data when permitted.
Preamble / Prompt Revealing (Non-Sensitive)	Extracting system prompts or preambles that do not contain sensitive, non-public information.
Extraction of Non-Sensitive / Public Information	Extraction of non-sensitive / public model information (e.g., model name or architectural approach, if publicly known).
Imprecise / Non-confidential Model Weight Extraction	Attacks in which the architecture and weights are not extracted precisely, or weights are extracted from a non-confidential model (e.g. Gemma).
Low-Impact Disruptions to User Experience	Minor disruptions like adding harmless text, minor formatting errors, easily ignored deviations from a user's typical experience.
Copyright issues	Content generation appearing to violate copyright. Google's generative AI products already include a dedicated reporting channel for these types of content issues, accessible via in-product links.

Google's generative AI products already have a dedicated reporting channel for these types of content issues. You can also report inappropriate content for legal reasons here.

Reward amounts for abuse-related vulnerabilities

Note: Rewards for abuse-related vulnerabilities range from USD $100 to $13,337. The reward amount for these abuse-related bugs depends on the potential probability and impact of the

		Impact [1]		
		High	**Medium**	**Low**
Probability [2]	**High**	Up to $13,337	$3,133.7 to $5,000	$1,337
	Medium	$3,133.7 to $5,000	$1,337	$100 to $500
	Low	$1,337	$100 to $500	HoF Credit

“[1] The impact assessment is based on the attack’s potential for causing privacy violations, financial loss, and other user harm, as well as the size of the user-base at risk.”

“[2] The probability assessment takes into account any prerequisites to conducting the attack —such as when the attacker needs to guess a victim's ID or needs prior access to the victim's device or account—and the likelihood of the exploitation being discovered by the victim.”

The final amount is always chosen at the discretion of the reward panel. In particular, we may decide to pay higher rewards for unusually clever or severe vulnerabilities; decide to pay lower rewards for vulnerabilities that require unusual user interaction; decide that a single report actually constitutes multiple bugs; or that multiple reports are so closely related that they only warrant a single reward.

We understand that some of you are not interested in money. We offer the option to donate your reward to an established charity. If you do so, we will double your donation - subject to our discretion. Any rewards that are unclaimed after 12 months will be donated to a charity of our choosing.

Investigating and reporting bugs

When investigating an abuse-related vulnerability, please, only ever target your own accounts. Never attempt to access anyone else's data and do not engage in any activity that would be disruptive or damaging to your fellow users or to Google.

If you have found an abuse-related vulnerability, please submit your report through our standard form (report to *Abuse VRP*). **Please be succinct**: your report is triaged by security engineers and a short proof-of-concept link is more valuable than a video explaining the consequences of the concrete case of potential abuse. If necessary, you can use this PGP key.

Note that we are only able to answer abuse-related vulnerability reports. Non-security/abuse bugs and queries about problems with your account should instead be directed to Google Help Centers.

Frequently asked questions

Q: My report has not been resolved within the first week of submission. Why hasn't it been resolved yet?

A: Reports that deal with potential abuse-related vulnerabilities may take longer to assess, because reviewing our current defense mechanisms requires investigating how a real life attack would take place and reviewing the impact and likelihood requires studying the type of motivations and incentives of abusers of the submitted attack scenario against one of our products.

Legal points

We are unable to receive reports and/or issue rewards to individuals or entities that are on sanctions lists, or who are in territories subject to sanctions (e.g., Cuba, Iran, North Korea, Syria, Crimea, and the so-called Donetsk People's Republic and Luhansk People's Republic). Additionally, due to administrative difficulties, we no longer issue rewards to individuals or entities located in Russia or Belarus.

You are responsible for any tax implications depending on your country of residency and citizenship. There may be additional restrictions on your ability to enter depending upon your local law.

or not to pay a reward has to be entirely at our discretion.

Of course, your testing must not violate any law, or disrupt or compromise any data that is not your own.

1 <!--showValues-- >

Rules

About This Section

Android & Friends

Chrome & Friends

Google & Friends

Open Source

Google Open Source Software Vulnerability Reward Program Rules

Open Source Security Subsidies Rules

OSS-Fuzz Reward Program Rules

Patch Rewards Program Rules

Tsunami Patch Rewards Program Rules

Other

Our Rewards Philosophy

Every week, a group of senior Googlers on our product security team meets to meticulously review and decide reward amounts for all recent bugs reported to us through our Google Vulnerability Reward Program. Based on the researcher's report and the initial triage of the bug by our team, the panel's task is to determine the impact of the given security issue, and to assign a consistent and fair reward amount to every notable find. This article is meant to shed light on how the panel decides rewards for the thousands of vulnerability reports it evaluates, and does not in any form alter the Google VRP rules.

For every reported vulnerability, the security impact is first of all evaluated by looking at the most dangerous attack scenario that the panel can come up with. We usually assign lower severity to problems that hinge on the existence of other, not-yet-discovered bugs to become exploitable; we do the same for findings that require extremely unlikely user interactions or other rarely-met prerequisites. On the flip side, when we can come up with higher-impact attack vectors that the original reporter hadn't considered in the submitted report, we bump up the score accordingly.

In a second step, our evaluation of the security impact then focuses on several additional factors:

- The type of exploit (RCE, XSS etc.), and in particular the kinds of malicious activities the exploit enables (command/SQL injection, sandbox escapes, data exfiltration etc.)
- The sensitivity of the affected application or service. For example, vulnerabilities affecting tier 0 and tier 1 applications as specified in our public list are considered higher impact and are thus eligible for higher rewards than issues affecting applications belonging to lower tiers. That said, if a report demonstrates how a bug in a lower-tier application can lead to an impact equivalent to compromising a higher-tier target (e.g. via some backend

Other aspects of the process worth noting include:

- Over time, we tend to adapt the reward amounts to provide balanced incentives for external researchers – especially as we find that certain classes of targets become more difficult to attack.
- When receiving multiple reports, we typically only reward once per root cause and group similar vulnerabilities together. For example, if there's a service that accidentally disabled CSRF protection, we wouldn't issue a reward for every handler that had CSRF protection disabled, but would instead issue a reward for the most serious CSRF vulnerability in the code.
- We reward exclusively based on security impact – i.e. rewards are not driven by PR, or by legal or reputational risks.
- When a report doesn't technically match the scope, or the impact isn't there, but we appreciate knowing about the issue, or the report led to a change in our products, we'll credit you on our Honorable Mentions board. Examples include bugs in recent acquisitions or bugs in apps that don't deal with user data.
- When in doubt about the exact impact of an issue, we side with researchers and reward the higher amount.
- In some cases, we'll decide to reward duplicate findings, e.g. if the duplicate report significantly contributes to improving the fix or allows us to fix the issue faster.
- A reward is only issued once a quorum of panel members have voted and reached agreement. This can occasionally result in delays as additional complex questions arise.

For more details on how reward amounts are determined, see the rules document for the Google VRP.

While the above description applies specifically to the Google VRP, the basics are the same for all other VRPs at Google: Based on an existing set of rules and an initial triage of the reported issue, a panel comes together to determine the issue's exact severity, and, on that basis, the exact amount that will be rewarded to the researcher who reported the issue.

And now, it's time for you to share your reports for scrutiny by our panel. Good luck bug hunting – we're looking forward to triaging and rewarding your report!

<!--showValues-->

Rules

About This Section

Android & Friends

Chrome & Friends

Google & Friends

Open Source

Other

Google and Alphabet Vulnerability Reward Program (VRP) Rules

Vulnerability Reward Program for Google-owned and Alphabet (Bet) subsidiary web properties, running continuously since November 2010.

Services in scope

In principle, any Google-owned or Alphabet (Bet) subsidiary web service that handles reasonably sensitive user data is intended to be in scope. This includes virtually all the content in the following domains:

- *.google.com
- *.youtube.com
- *.blogger.com
- *.deepmind.com
- *.waymo.com
- *.wing.com

Bugs in Google- and Waymo-developed apps published in the Apple App Store will also qualify. For information on further services and devices that are in scope of different reward programs, see the rules for the following programs:

- Abuse Vulnerability Reward Program Rules
- Android and Google Devices Security Reward Program Rules
- Chrome Vulnerability Reward Program Rules
- ChromeOS Vulnerability Reward Program Rules
- Cloud Vulnerability Reward Program Rules
- Google Mobile Vulnerability Reward Program Rules
- Google Open Source Software Vulnerability Reward Program Rules
- Chrome Extensions Vulnerability Reward Program Rules
- Verily Bug Bounty Program Rules on HackerOne

- **Third-party websites** – Some Google-branded services hosted in less common domains may be operated by our vendors or partners. We can't authorize you to test these systems on behalf of their owners and will not reward such reports. Please read the fine print on the page and examine domain and IP WHOIS records to confirm. If in doubt, talk to us first!
- **Recent acquisitions** – To allow time for internal review and remediation, newly acquired companies are subject to a six-month blackout period. Bugs reported sooner than that will typically not qualify for a reward.

Qualifying vulnerabilities

Any design or implementation issue that substantially affects the confidentiality or integrity of user data is likely to be in scope for the program. Common examples include:

- Cross-site scripting,
- Cross-site request forgery,
- Mixed-content scripts,
- Authentication or authorization flaws,
- Server-side code execution bugs,
- XSLeak bugs.

Note that the scope of the program is limited to technical vulnerabilities in Google-owned browser extensions, mobile, and web applications; please do not try to sneak into Google offices, attempt phishing attacks against our employees, and so on.

Out of concern for the availability of our services to all users, please do not attempt to carry out DoS attacks, leverage black hat SEO techniques, spam people, or do other similarly questionable things. We also discourage the use of any vulnerability testing tools that automatically generate very significant volumes of traffic.

Non-qualifying vulnerabilities

Depending on their impact, some of the reported issues may not qualify. Although we review them on a case-by-case basis, here are some of the common low-risk issues that typically do not earn a monetary reward:

- **Vulnerabilities in *.bc.googleusercontent.com or *.appspot.com.** These domains are used to host applications that belong to Google Cloud customers. The Vulnerability Reward Program does not authorize the testing of Google Cloud customer applications. Google Cloud customers can authorize the penetration testing of their own applications (*read more*), but testing of these domains is not within the scope of or authorized by the Vulnerability Reward Program.
- **Cross-site scripting vulnerabilities in "sandbox" domains** (*read more*.) We maintain a number of domains that leverage the same-origin policy to safely isolate certain types of untrusted content; the most prominent example of this is *.googleusercontent.com. Unless an impact on sensitive user data can be demonstrated, we do not consider the ability to execute JavaScript in that domain to be a bug.
- **Execution of owner-supplied JavaScript in Blogger.** Blogs hosted in *.blogspot.com are no different from any third-party website on the Internet. For your safety, we employ spam and malware detection tools, but we do not consider the ability to embed JavaScript within your own blog to be a security bug.
- **URL redirection** (*read more*.) We recognize that the address bar is the only reliable security indicator in modern browsers; consequently, we hold that the usability and security benefits of a small number of well-designed and closely monitored redirectors outweigh their true risks.
- **Legitimate content proxying and framing.** We expect our services to unambiguously label third-party content and to perform a number of abuse-detection checks, but as with redirectors, we think that the value of products such as Google Translate outweighs the risk.
- **Bugs requiring exceedingly unlikely user interaction.** For example, a cross-site scripting flaw that requires the victim to manually type in an XSS payload into Google Maps and then double-click an error message may realistically not meet the bar.
- **Logout cross-site request forgery** (*read more*.) For better or worse, the design of HTTP cookies means that no single website can prevent its users from being logged out; consequently, application-specific ways of achieving this goal will likely not qualify. You may be interested in personal blog posts from Chris Evans and Michal Zalewski for more background.
- **Flaws affecting the users of out-of-date browsers and plugins.** The security model of the web is constantly being fine-tuned. The panel typically does not reward reports that describe issues that affect only the users of outdated or unpatched browsers.

security risk, please let us know.

- **Email spoofing on Gmail and Google Groups.** We are aware of the risk presented by spoofed messages and are taking steps to ensure that the Gmail filter can effectively deal with such attacks.
- **User enumeration.** Reports outlining user enumeration are not within scope unless you can demonstrate that we don't have any rate limits in place to protect our users.
- **Bypassing the limit of accounts that can be verified with a given SMS number.** We often receive reports about users being able to bypass our SMS limit for verifying accounts. There are actually two different quotas per number for account verification, one via 'SMS' and a different one via 'Call Me'.

Monetary rewards aside, vulnerability reporters who work with us to resolve security bugs in our products will be credited on the Leaderboard. If we file an internal security bug, we will acknowledge your contribution on that page.

Reward amounts for security vulnerabilities

The following table outlines the standard rewards for the most common classes of bugs, and the sections that follow it describe how these rewards can be adjusted to take into account characteristics such as report quality and other factors. To understand more about our general approach to vulnerability rewards, you can read our dedicated article.

Category	Examples	Google applications on Tier 0 domains [0], or global impact [1] (T0)	Google applications on Tier 1 domains [2] (T1)	Normal Google Applications (T2) Examples include: *.google.com, *.youtube.com, *.blogger.com, *.admob.com	Applica on acquis Tier 0, ' domair [4] (T

Remote code execution (S0)	Command injection, deserialization bugs, sandbox escapes	$101,010	$101,010	$75,000	$10,000
Unrestricted file system or database access (S1)	Unsandboxed XXE, SQL injection	$75,000	$75,000	$50,000	$10,000
Logic flaw bugs leaking or bypassing significant security controls impacting SPII [5] (S2a)	SPII – Direct object reference, remote user impersonation	$50,000	$50,000	$31,337	$5,000
Logic flaw bugs leaking or bypassing significant security controls impacting PII or other user confidential information (S2b)	PII or other user confidential information – Direct object reference, remote user impersonation	$31,337	$20,000	$13,337	$2,500
Logic flaw bugs leaking or bypassing significant security controls impacting other data/systems (S2c)	Other – Direct object reference, remote user impersonation	$13,337	$10,000	$5,000	$1,337
Vulnerabilities giving access to client or authenticated session of the logged-in victim					

on the client (C0)	Mobile / Hardware: Code execution	$20,000	$15,000	$10,000	$500
Other valid security vulnerabilities (C1)	Web: CSRF, Clickjacking, XSLeaks Mobile / Hardware: Information leak, privilege escalation	$5,000 - $15,000	$5,000 - $15,000	$1,337- $10,000	$500

[0] Tier 0 domains are defined as domains where a critical vulnerability (e.g. XSS or authorization bypass) could lead to a compromise of a user's account or execution of code on their system.

[1] “Global impact” refers to any vulnerability that impacts a significant portion of the Internet, through integration with a product in Google VRP scope. For example, an XSS in Google Analytics embedded JavaScript.

[2] Tier 1 domains are defined as domains where a vulnerability could disclose particularly sensitive user data.

[3] Tier 0 and tier 1 acquisition domains are defined as domains where a compromise may lead to access of highly-sensitive information.

[4] Note that acquisitions qualify for a reward only after the initial six-month blackout period has elapsed.

[5] SPII is defined here.

The final amount is always chosen at the discretion of the reward panel. We understand that some of you are not interested in money. We offer the option to donate your reward to an established charity. If you do so, we will double your donation – subject to our discretion. Any rewards that are unclaimed after 12 months will be donated to a charity of our choosing.

Rewards are adjusted based on the quality of the report. The main factors considered are:

- Demonstrated security impact of the reported vulnerability – Impact is judged based on the **actual reported impact** of the vulnerability, and not on a **potential impact** of the vulnerability. For example, reports related to API keys are often not accepted without a valid attack scenario (see Understanding API key leaks). Another example would be to make sure to fully demonstrate the impact of an XSS vulnerability (see When reporting XSS, don't use alert(1)).
- Overall report quality – See the following sections for details.

Exceptional Quality (1.5x reward amount)

An Exceptional Quality report has both of the following characteristics (in addition to the characteristics of a Good Quality report described below):

- Includes a proposed patch or effective mitigation to the vulnerability
- Includes a root cause analysis, which helps us find other similar variants of the issue

Good Quality (1x reward amount)

A Good Quality report has several of the following characteristics:

- An accurate and detailed description of the issue including any relevant version numbers for applications, OS, web browsers, hardware device models etc.
- A proof-of-concept that effectively, quickly, and easily demonstrates the vulnerability with any applicable reproduction output (e.g., video recording, APK file, etc.)
- A step-by-step explanation on how to reliably reproduce the vulnerability
- A succinct analysis and demonstration of the impact of the vulnerability

In addition to the characteristics listed above, we also expect the researcher to be responsive when asked questions and accurately answer any queries about the vulnerability

Low Quality (0.5x reward amount)

Common Precedents / Upgrades and Downgrades

While we try to account for everything, our table above is not perfect and cannot cover all possible situations. In complex situations, our rewards panel can still reward submissions, but will apply modifiers to the reward amount.

Precedents

Type	Amount	Reasoning
Incidents	$500	Security issues that are not technical vulnerabilities. For example, a Googler accidentally sharing a confidential document publicly.
3rd-party system	$100-$500	Standard amount for an issue with a 3rd-party integration or configuration where Google controls the integration or configuration, but not the underlying application.
Documentation update (security impacting)	$100-$500	Standard amount for reports of incorrect documentation which is likely to lead to insecure states for users.
Documentation update	HOF credit	Standard for reports of insufficient or incorrect documentation.

Upgrades

Type	Amount	Reasoning
Panel discretion	+$1k	A standard bonus the panel can issue at their discretion.
Time-limited bonus	+75%	Occasionally, we will publish time-limited bonuses for specific VRP targets.
Swag	+???	It's a mystery how to receive this upgrade.

The following table describes potential reasons for downgrades. A downgrade means your reward will be reduced by a predetermined step, for example $5000 will always step down to $3,133.7. Multiple downgrades can be applied.

Category	Reasoning
Impact	Downgrade applied for minor impact.
Prior access	Downgrade applied because the attacker needs to have prior access to an impacted victim's resource.
Project access	Downgrade applied because the attacker needs to have access to an impacted victim's project.
User interaction	Downgrade applied because the attack requires significant user interaction by the victim.
Exploitability	Downgrade applied because the attack is still possible, but we consider the vulnerability unexploitable in practice. Reports of this kind may also receive a reduced quality rating or not be accepted during triage depending on how likely an issue is to affect another user.

Examples

Here are a few examples showing how the rules described above result in reward amounts:

Bug Description	Reward
An RCE in a server serving accounts.google.com, report is of exceptional quality (1.5x).	(T0, S0) $101,010 * 1.5 = **$151,515**
An IDOR on a `*.google.com` domain which can be used to exfil a user's saved address (PII) to a third party, report is of normal quality (1x).	(T2, S2b) $13,337 * 1.0 = **$13,337**
An easy-to-exploit XSS which impacts all pages translated to English via Google Translate ("global impact" tier), report is of normal quality (1x).	(T0, C0) $20,000 * 1.0 = **$20,000**

hours, report is of normal quality (1x).	$7,500 * 1.0 = **$7,500**
An XSS on a Tier 0 acquisition domain that is only exploitable after a million HTTP requests, report is of exceptional quality (1.5x).	(T3a, C1, with exploitability downgrade) $200 * 1.5 = **$300**

Investigating and reporting bugs

When investigating a vulnerability, please, only ever target your own accounts. Never attempt to access anyone else's data and do not engage in any activity that would be disruptive or damaging to your fellow users or to Google.

Note: Visit our Bug Hunter University articles to learn more about sending good vulnerability reports.

If you have found a security vulnerability, please submit your report through the form available on our report page. **Please be succinct**: your report is triaged by security engineers and a short proof-of-concept link is more valuable than a video explaining the consequences of an XSS bug. If necessary, you can use this PGP key.

Note that we are only able to answer to technical vulnerability reports. Non-security bugs and queries about problems with your account should be instead directed to Google Help Centers.

Frequently asked questions

Q: What if I found a vulnerability, but I don't know how to exploit it?

A: We expect that vulnerability reports sent to us have a valid attack scenario to qualify for a reward, and we consider this to be a critical element of vulnerability research. Reward amounts are decided based on the maximum impact of the vulnerability, and the panel is willing to reconsider a reward amount, based on new information (such as a chain of bugs, or a revised attack scenario).

Q: How do I demonstrate the severity of the bug if I'm not supposed to snoop around?

pay higher rewards for otherwise well-written and useful submissions where the reporter didn't notice or couldn't fully analyze the impact of a particular flaw.

Q: I found an outdated software (e.g. Apache or Wordpress). Does this qualify for a reward?

A: Please perform due diligence: confirm that the discovered software had any noteworthy vulnerabilities, and explain why you suspect that these features may be exposed and may pose a risk in our specific use. Reports that do not include this information will typically not qualify.

Q: What happens if I disclose the bug publicly before you had a chance to fix it?

A: Please read our stance on coordinated disclosure. In essence, our pledge to you is to respond promptly and fix bugs in a sensible timeframe - and in exchange, we ask for a reasonable advance notice. Reports that go against this principle will usually not qualify, but we will evaluate them on a case-by-case basis.

Q: I wish to report an issue through a vulnerability broker. Will my report still qualify for a reward?

A: We believe that it is against the spirit of the program to privately disclose the flaw to third parties for purposes other than actually fixing the bug. Consequently, such reports will typically not qualify.

Q: What if somebody else also found the same bug?

A: First in, best dressed. You will qualify for a reward only if you were the first person to alert us to a previously unknown flaw.

Q: My employer / boyfriend / dog frowns upon my security research. Can I report a problem privately?

A: Sure. If you are selected as a recipient of a reward, and if you accept, we will need your contact details to process the payment. But if the **Make my profile public** option in your profile is not selected, there won't be any public mention of you on our Leaderboard.

A: The dashboard for the participants in Google's VRP program. It dynamically creates the 0x0A, Leaderboard, and Honorable Mentions lists.

Q: Is my profile data publicly available?

A: Only if you select the **Make my profile public** option in your profile. If you select this option, the data the profile holds is used on our Leaderboard, i.e. on the 0x0A, Leaderboard, and Honorable Mentions lists.

Q: How is the Honorable Mentions list sorted?

A: The list is sorted based on the volume of valid bug submissions, the ratio of valid vs. invalid submissions, and the severity of those submissions.

Q: My account was disabled after doing some tests. How can I get my account restored?

A: We recommend that you create an account dedicated only to testing before beginning any tests on our products, since we cannot guarantee that you will get access back to your account if it is disabled due to your testing activities. If you accidentally used a non-test account or you suspect your personal account was disabled due to your testing, you can request to have your account restored by Signing in to your Google Account and selecting Try to Restore.

Legal points

We are unable to receive reports and/or issue rewards to individuals or entities that are on sanctions lists, or who are in territories subject to sanctions (e.g., Cuba, Iran, North Korea, Syria, Crimea, and the so-called Donetsk People's Republic and Luhansk People's Republic). Additionally, due to administrative difficulties, we no longer issue rewards to individuals or entities located in Russia or Belarus.

You are responsible for any tax implications depending on your country of residency and citizenship. There may be additional restrictions on your ability to enter depending upon your local law.

or not to pay a reward has to be entirely at our discretion.

Of course, your testing must not violate any law, or disrupt or compromise any data that is not your own.